JOURNAL COMPUTER SCIENCES INSTITUTE

JCSI 7 (2018) 108-115
Wysłane: 2017-11-19
Przyjęte: 2017-11-24

Analiza wydajnościowa i możliwościowa narzędzia Laravel do tworzenia nowoczesnych aplikacji webowych

Przemysław Mincewicz*, Małgorzata Plechawska-Wójcik
Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Tematyką badaną w artykule jest sprawdzenie możliwości narzędzia do tworzenia aplikacji webowych Laravel. Przedstawiono przegląd literatury traktującej o Laravel, jego najpopularniejsze moduły oraz dodatkowe możliwości. Na podstawie stworzonej aplikacji testowej napisanej przy pomocy trzech narzędzi – Laravel, Symfony oraz Zend porównano je ze sobą pod względem wydajności oraz możliwości. Aplikacją testową była baza albumów muzycznych o trzech zestawach danych – z jednym, dwoma oraz pięcioma tysiącami albumów. Badanie wydajności narzędzia Laravel na tle Zend Framework i Symfony sporządzono na podstawie między innymi czasu ładowania danych z bazy oraz ich wyszukiwania w kolekcji danych. Podsumowanie informacji zawartych w artykule stwierdza, że Laravel do narzędzie nastawione na komfort pisania aplikacji, odstające jednak na tle innych narzędzi wydajnością.

Słowa kluczowe: framework Laravel; język PHP; MVC; tworzenie aplikacji w narzędziu Laravel; porównanie narzędzi języka PHP

*Autor do korespondencji.

Adres e-mail: p.mincewicz@gmail.com

Performance and possibility analysis of Laravel tool dedicated to create modern web applications

Przemysław Mincewicz*, Małgorzata Plechawska-Wójcik
[a] Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. The aim of this paper is verification of possibilities of Laravel PHP framework. An overview of literature is presented, which shows Laravel as a research tool in various scientific fields and compares it with other PHP language tools. The project of a test application was presented, which is a database of music albums with three datasets - one, two and five thousand albums. Laravel's Zend Framework and Symfony performance survey was based on, among other things, database loading time, database searches, and the time to generate a single album view with relationships. Summary of the information contained in the paper states that Laravel is an easy-to-use tool, with a number of app creation aids, but with performance outweighed by the frameworks compared to it, the most efficient of which is Zend Framework.

Keywords: Laravel framework; PHP programming language; MVC; Laravel application development; PHP frameworks comparison

*Corresponding author.

E-mail address: p.mincewicz@gmail.com

1. Wprowadzenie

Aplikacje internetowe są w dzisiejszych czasach postrzegane w taki sam sposób jak aplikacje desktopowe. Wymaga się od nich jednakowych funkcjonalności, przystępności dla użytkownika, aktualizacji rozwojowych oraz ciągłego wsparcia technicznego. Zaletą wyróżniającą aplikacje webowe jest dostępność oraz uniwersalność – do uruchomienia ich wystarczy jedynie przeglądarka internetowa.

Dla deweloperów obecny rynek narzędzi do tworzenia internetowych aplikacji jest niezwykle bogaty. Wiele języków programowania pozwala na tworzenie w pełni funcjonalnych i rozbudowanych aplikacji, a każdy z nich zawiera co najmniej kilka narzędzi wspomagających ich pisanie. W artykule zwrócono szczególną uwagę na nowe i dynamicznie rozwijające się narzędzie Laravel, które oparte jest w głównym stopniu o język PHP. Bardzo szybko zyskał on wielkie grono zwolenników i stał się jednym z najpopularniejszych narzędzi do tworzenia nowoczesnych aplikacji internetowych.

1.1. Charakterystyka narzędzia Laravel

Wraz z rozwojem internetu oraz powstawaniem nowych urządzeń, które mogły z niego korzystać, w procesie tworzenia aplikacji internetowych większą wagę przykładano do skalowalności oraz większej uniwersalności programowania wspomagającej multiplatformowość. Język PHP, będącym początkowo jedynie narzędziem tworzenia spersonalizowanych stron internetowych [8], wkomponował się idealnie w zaistniałą sytuację i stał się jednym z najważniejszych języków programowania nowoczesnych aplikacji internetowych.

Wraz z rozwojem narzędzi do tworzenia aplikacji w języku PHP, takich jak CodeIgniter, CakePHP, Symfony czy Laravel zmieniał się sposób implementacji odpowiednich warstw w kodzie poprzez zróżnicowane technologie obsługi baz danych czy tworzenia widoków dla użytkownika. Dużą uwagę zwrócono na bezpieczeństwo użytkownika. W okresie od połowy 2006 do 2010 roku aplikacje PHP zwiększyły swoje bezpieczeństwo o blisko 9% [1] poprzez zmiany wyłącznie

w strukturze tworzenia aplikacji i zmianom w CORE (podstawie) języka.

1.2. Przegląd literatury

Laravel jako narzędzie do tworzenia aplikacji internetowych w bardzo krótkim czasie zyskał wielu sympatyków. Złożyło się na to wiele czynników, między innymi prosta implementacja, bogata społeczność deweloperska oraz ciągłe wsparcie najnowszych rozwiązań języka PHP. Badania przeprowadzone przez portal sitepoint ukazują, że Laravel w ciągu 4 lat stał się najpopularniejszym narzędziem języka PHP, dwukrotnie częściej wykorzystywanym niż CodeIgniter i Symfony.

W niewielkim okresie czasu od powstania Laravel był porównywany z najpopularniejszymi narzędziami tworzenia aplikacji modelem MVC (ang. Model-View-Controller), takich jak Symfony, Yii czy CakePHP. W badaniach przeprowadzonych przez pana Adriana Zurkiewicza oraz Marka Miłosza, którzy stworzyli w pełni funkcjonalną platformę edukacyjną w wyżej wymienionych narzędziach, nie okazuje się on jednak narzędziem idealnym [2]. Wśród aspektów branych pod uwagę podczas analizy, z których wyróżnić należy obsługę baz danych, techniki programowania skupiające się na najbardziej efektywnym pisaniu kodu oraz realnej pracy programisty, Laravel okazał się liderem jedynie we wsparciu dla technologii. Ukazało to jego otwartość i wiele możliwości rozwiązywania problemów, jednak odsłoniło braki, z których szczególnie poważnym jest bardzo mała efektywność działania. Zestawienie wszystkich analiz wykazało, że Laravel nie jest najlepszym rozwiązaniem dla projektów rozbudowanych aplikacji MVC umożliwiających wielopoziomową komunikację między jej użytkownikam.

Ze względu na wysoką dostępność Laravel jest bardzo popularnym narzędziem w aplikacjach badawczych. Jedną z nich jest system sterowania domowymi urządzeniami, monitorujący jednocześnie zużycie energii [3]. Głównym jego założeniem jest obniżenie całkowitego zużycia energii urządzeń poprzez dynamiczne i automatyczne wyłączanie niewykorzystywanych. Dzięki takiemu projektowi użytkownik ma możliwość własnoręcznego zarządzania wszystkimi urządzeniami zsynchronizowanymi z systemem, wglądu w raporty odczytów oraz dostosowanie opcji wspierających oszczędzanie energii do własnych potrzeb. Po przeprowadzeniu badań i analizie raportów zmniejszono zużycie energii ze wszystkich branych pod uwagę urządzeń o blisko 59%.

Inną bardzo ciekawą aplikacją badawczą jest InverPep [9], traktującą o tematyce biologicznej. Stworzona przez Universidad Nacional de Colombia jest bazą peptydów antybakteryjnych (AMP), czyli podstawowych mechanizmów obronnych wytwarzanych przez rośliny i zwierzęta [4]. Narzędzie Laravel, poza utworzeniem bazy oraz strony wizualnej, posiada zaimplementowany algorytm obliczania właściwości fizykochemicznych wielu peptydów jednocześnie - CALCAMPI [5]. Każdy AMP jest opisany źródłem, właściwościami fizykochemicznymi, strukturą oraz aktywnością biologiczną.

Poza aplikacjami typowo badawczymi, gdzie narzędzie w gruncie rzeczy pełni drugorzędną rolę, Laravel wykorzystywany jest w aplikacjach testujących działanie innowacyjnych rozwiązań algorytmicznych oraz programistycznych. Przykładem takiej aplikacji jest zaproponowane przez He Ren Yu wdrożenie narzędzia Laravel bez wykorzystywania tradycyjnego wzorca MVC - programista skupia się jedynie na warstwie logiki [6]. Ma to na celu ujednolicenie procesu rozwoju aplikacji, lepszej implementacji założeń klienta oraz łatwiejszym zarządzaniem danymi, które wprowadzanie są do projektu poprzez plik XML. Takie rozwiązanie tworzenia aplikacji za pomocą Laravel zestawiono z tradycyjnym MVC z wykorzystaniem narzędzia CodeIgniter. Wynikiem okazało się dużo efektywniejsze tworzenie aplikacji za pomocą Laravel, co może świadczyć jednocześnie o łatwiejszym programowaniem w tym narzędziu jak i skuteczności prezentowanego przez autora rozwiązania.

Laravel nie jest narzędziem idealnym dla wszystkich aplikacji badawczych. Przykładem bazującej na wzorcu MVC jest system monitorujący i badający wpływ czynników naturalnych na wydajność technologii informacyjnych [7]. Głównym celem jej działania jest badanie parametrów środowiskowych takich jak temperatura i wilgotność powietrza pobranych przez przez sieć Sentinel i generowanie na ich podstawie szczegółowych raportów. Laravel jako narzędzie nie posiada bogatego wsparcia dla europejskiej sieci naukowej, brak w nim wielu przydatnych funkcji, między innymi uwierzytelniania z siecią. Przeprowadzone badania aplikacji na wielu przeglądarkach ukazało, że Laravel w jednakowy sposób jest przez każdą z nich interpretowany. Nie zmienia to faktu, że tak rozbudowana i niezwykle istotna technologia jaką jest sieć Sentinel nie dostała jeszcze wsparcia ze strony narzędzia Laravel.

1.3. Przedmiot i cel badań

Badania przeprowadzono w celu sprawdzenia wydajności i możliwości aplikacji napisanych w narzędziu Laravel. Jako punkty odniesienia posłużyły dwa popularne frameworki PHP – Symfony, czyli jeden z najpopularniejszych oraz Zend – w wielu badaniach uznawany za najwydajniejsze narzędzie języka PHP. Dadzą one pogląd na to co odpowiada za popularność Laravel - wydajność czy możliwości.

Przedmiotem badań jest jest aplikacja testowa. Badania zostaną przeprowadzone jako porównanie jej wersji w trzech narzędziach pod względem:

- wydajności:
  - czas dostępu i odczyt danych z serwera dla różnych zbiorów danych – wykorzystane zostaną 3 bazy, odpowiednio z 1 tys. albumów, 2 tys. albumów oraz 5 tys.
  - czas generowania strony przez narzędzie,
  - obsługa plików graficznych, czy występuje cachowanie,
- możliwości, czyli wykorzystanie wbudowanych w narzędzie mechanizmów tworzenia formularzy, widoków list obsługi bazy danych.
  - czas dostępu i odczyt danych z serwera dla różnych zbiorów danych – wykorzystane zostaną 3 bazy, odpowiednio z 1 tys. albumów, 2 tys. albumów oraz 5 tys.
  - czas generowania strony przez narzędzie,
  - obsługa plików graficznych, czy występuje cachowanie,
  - obsługa kolekcji obiektów pobieranych z bazy danych,

Aplikacje różnią się od siebie jedynie narzędziem, w którym zostały stworzone – bazy danych oraz ogólne funkcjonalności są takie same. Wyniki badań pozwolą przedstawić, które z wykorzystanych w pracy narzędzi jest najbardziej efektywne w jednakowych warunkach działania oraz poziom złożoności tworzenia jednakowej aplikacji.

2. Projekt aplikacji testowej

W celu analizy wydajności oraz możliwości narzędzia Laravel na tle innych narzędzi języka PHP stworzone zostaną trzy jednakowe aplikacje internetowe – każda z wykorzystaniem innego narzędzia – Laravel 5, Symfony 3 oraz Zend 3. Oparte będą o wzorzec MVC oraz indywidualne dla każdego z nich mechanizmy obsługi baz danych MySQL. Tematem aplikacji będzie zbiór informacji o albumach muzycznych, ich wykonawcach, wydawcach oraz zawartych w nich utworach.

Aplikacja będzie posiadać podstawowe funkcjonalności aplikacji internetowej wzbogacone o dodatkowe możliwości, takie jak:

- dodawanie albumów, wykonawców i wydawców,
- dodawanie okładek do albumów,
- powiązanie albumu z wykonawcą i wydawcą,
- dodawanie utworów do albumów,
- edycję albumów, wykonawców, wydawców i utworów,
- usuwanie albumów, wykonawców, wydawców i utworów,
- wyświetlanie listy albumów, wykonawców i wydawców,
- wyświetlanie szczegółowych informacji o albumach, w tym listę utworów, wykonawcach i wydawcach,
- filtrowanie wybranych pól na liście albumów, wykonawców oraz wydawców ,
- wyszukiwanie pozycji na liście albumów, wykonawców i wydawców w wybranym kryterium wpisanym w pole tekstowe,
- rejestracja nowego użytkownika i system logowania do aplikacji.

Aplikacja zapewnia także bezpieczeństwo użytkowania, poprzez obsługę sesji i ciasteczek, jest responsywna (przystosowana do wielu typów urządzeń) Umożliwia użytkownikowi filtrowanie oraz wyszukiwanie informacji o albumach muzycznych, ich wykonawcach oraz wydawcach, wymusza od użytkownika zalogowanie do aplikacji – w przypadku braku danych do logowania wymagana jest rejestracja.

Dodawanie nowego albumu muzycznego wymaga wcześniej definicji artystów i wydawców – relacje między tymi elementami są wymagane i widoczne w szczegółach płyty. Utwory z kolei dodawane są wyłącznie z poziomu szczegółów albumu – nie posiadają tym samym własnego, wyodrębnionego widoku, stale przypisane do konkretnej kompozycji.

2.1. Kryteria analizy aplikacji

Badania podzielone zostaną na dwa etapy:

1) Analizę możliwości narzędzi, które przeprowadzono podczas tworzenia aplikacji. W skład badania wchodzą:
   a. mechanizm tworzenia kontrolerów,
   b. mechanizm tworzenia widoków,
   c. mechanizm tworzenia formularzy,
   d. mechanizm obsługi bazy danych,
   e. mechanizm obsługi kolekcji obiektów (ORM),
   f. mechanizm migracji bazy danych,
   g. łatwość budowy aplikacji - liczbę linii kodu, jakie należy przygotować do uzyskania określonego efektu.
2) Analizę ogólnej wydajności działania narzędzi, które przeprowadzono na gotowych aplikacjach. W skład badania wchodzą:
   a. czas dostępu do bazy danych z 1 tys. albumów,
   b. czas odczytu z bazy danych z 1 tys. albumów,
   c. czas wyszukiwania pojedynczego albumu w bazie danych z 1 tys. albumów,
   d. czas wyszukiwania odpowiednich elementów w pobranej kolekcji danych w bazie z 1 tysiącem albumów,
   e. czas filtrowania albumów w bazie danych z 1 tys. albumów,
   f. czas filtrowania elementów w pobranej kolekcji danych w bazie z 1 tys. albumów,
   g. czas generowania kolekcji danych złożonych z wielu encji w bazie z 1 tys. albumów,
   h. czas dostępu do bazy danych z 2 tys. albumów,
   i. czas odczytu z bazy danych z 2 tys. albumów,
   j. czas wyszukiwania pojedynczego albumu w bazie danych z 2 tys. albumów,
   k. czas wyszukiwania elementów w pobranej kolekcji danych w bazie z 2 tysiącami albumów,
   l. czas filtrowania albumów w bazie danych z 2 tys. albumów,
   m. czas filtrowania odpowiednich elementów w pobranej kolekcji danych w bazie z 2 tys. albumów,
   n. czas generowania kolekcji danych złożonych z wielu encji w bazie z 2 tys. albumów,
   o. czas dostępu do bazy danych z 5 tys. albumów,
   p. czas odczytu z bazy danych z 5 tys. albumów,
   q. czas wyszukiwania pojedynczego albumu w bazie danych z 5 tys. albumów,
   r. czas wyszukiwania pojedynczego albumu w bazie danych z 5 tys. albumów,
   s. czas filtrowania albumów w bazie danych z 5 tys. albumów,
   t. czas filtrowania elementów w pobranej kolekcji danych w bazie z 5 tys. albumów,
   u. czas generowania kolekcji danych złożonych z wielu encji w bazie z 5 tys. albumów,
   v. czas generowania strony,
   w. czas ładowania plików graficznych.

3. Najpopularniejsze biblioteki

Architektura narzędzia Laravel została oparta o moduły. Początkowo spotkało się to z dozą krytyki, jednak w ogromnym stopniu przyczyniło do powstania ogromnej społeczności deweloperów wokół środowiska. Mają oni w łatwy sposób możliwość dzielenia się swoimi rozwiązaniami wielu problemów związanych z programowaniem, dzięki czemu każdy może wykorzystać przygotowany przez nich dodatek. Instalacja takiego dodatku przebiega poprzez narzędzie Composer.

W dalszej części artykułu przedstawiono cztery najczęściej wykorzystywane dodatki w aplikacjach Laravel.

3.1. Laravel Collectives – Forms & HTML

Najpopularniejszym i najczęściej wykorzystywanym dodatkiem narzędzia Laravel jest Form & HTML. Automatyzuje i znacznie ułatwia on tworzenie widoków w aplikacji oraz formularzy, które są nieodzowną jej częścią - każde wysłanie żądania dla zapisu czy odczytu danych jest przeprowadzane poprzez formularz. Dodatek ten wydaje się idealnym dla każdego dewelopera - początkującego, nie chcącego tracić czasu bądź nie potrafiącego tworzyć rozbudowanych formularzy, czy też doświadczonego, który na bazie dodatku zbuduje własne, bardziej spersonalizowane formularze [10].

3.2. Laravel Debugbar

Dodatek Laravel Debugbar służy do integracji bardzo przydatnego deweloperom PHP narzędzia do debugowania - PHP Debug Bar z aplikacją Laravel. W dużym stopniu ułatwia on implementację narzędzia w aplikacji, która sprowadza się do standardowej instalacji dodatku, bez konieczności dodatkowych "ingerencji" w kod [11]. Dodatkowo narzędzie Debug Bar zostało wzbogacone o obsługę przekierowania JQuery i żądania Ajax oraz niestandardowych kolekcji danych, specyficznych dla aplikacji Laravel.

3.3. Laravel Excel

Dodatek Laravel Excel pozwala aplikacji na obsługę arkuszy kalkulacyjnych w formacie .xls oraz .csv. Pozwala na odczytywanie danych z plików automatycznie przekształcając je w postać zbioru obiektów jak i również na zapisywanie zbiorów do plików jako wykonywalne w arkuszu kalkulacyjnym [12].

3.4. Agent

Dodatek Agent przystosowany dla narzędzia Laravel służy do wykrywania platformy, z jakiej korzysta użytkownik aplikacji. Za jego pomocą można stwierdzić z jakiego urządzenia korzysta (komputer, smartfon, tablet), systemu operacyjnego oraz przeglądarki internetowej [13].

4. Dodatkowe możliwości narzędzia Laravel

W dalszej części artykułu przedstawiono dodatkowe możliwości oferowane przez narzędzie Laravel. Żadna z nich nie została wykorzystana podczas tworzenia aplikacji testowej – wyłączając REST API są to aspekty wyróżniające Laravel na tle innych frameworków, zatem założenie stworzenia identycznej aplikacji w trzech narzędziach byłoby niemożliwe.

REST API jest możliwością, którą oferują zarówno Laravel, Symfony oraz Zend, dlatego też służyła jako dodatkowy aspekt porównawczy.

4.1. REST API

Laravel, podobnie jak Symfony i Zend, pozwala na stworzenie komunikacji pomiędzy aplikacją klienta a danymi bez konieczności implementacji do niej pełnej funkcjonalności za nią odpowiadającą. Służy do tego REST API, czyli interfejs komunikacji pomiędzy klientem a serwerem.

Komunikacja aplikacji z API w dużym skrócie przebiega poprzez wysyłanie odpowiednich ścieżek, które odpowiednio interpretowane zwracają żądane dane lub wykonują odpowiednie akcje. Dane zwracane są w najczęściej w formacie .json, interpretowanym przez każdy język programowania.

Tworzenie REST API w narzędziu Laravel nie odbiega znacząco od standardowego, webowego projektu. Jedyna istotną różnicą jest specyficzne kierowanie zapytań (ang. routing) do odpowiednich metod w kodzie aplikacji.

4.2. Wykorzystanie chmury Amazon S3

Amazon Simple Storage Service (w skrócie Amazon S3) jest to internetowa przestrzeń gromadzenia, przechowywania oraz analizy danych. Dane przechowywane w taki sposób mogą być w jednakowy sposób wykorzystywane przez różne systemu i aplikacje – zarówno internetowe jak i mobilne oraz desktopowe.

Narzędzie Laravel, ze względu na rosnącą popularność oraz skuteczność wykorzystywania Amazon S3 posiada wbudowany mechanizm pozwalający w łatwy sposób wykorzystywać przestrzeń chmurową Amazon.

Wykorzystywanie takiej przestrzeni w aplikacjach internetowych wydaje się kolejnym krokiem w rozwoju języka PHP. Zmniejsza to ilość miejsca wykorzystywaną fizycznie przez aplikację oraz pozwala wykorzystywać te same dane w kilku aplikacjach, działających również równolegle – dane mogą być na bieżąco i w tym samym momencie nadpisywane/odczytywane.

4.3. Wykorzystanie bazy GeoIP

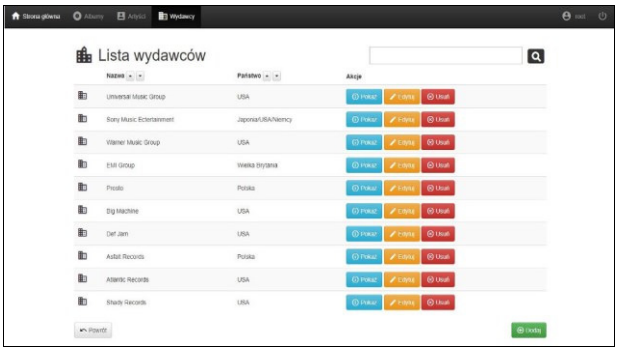
Określenie lokalizacji użytkownika korzystającego z aplikacji jest obecnie bardzo powszechne. Baza GeoIP jest to ogólnodostępny zbiór lokalizacji rozwijany przez firmę MaxMind. Dzięki takiej bazie mamy możliwość w bardzo dokładny sposób wyznaczyć lokalizację użytkownika poprzez jego adres IP.

Narzędzie Laravel posiada przystosowany dla siebie moduł korzystający z bazy GeoIP, rozwijany przez firmę Lyften. Jego implementacja w aplikacji nie odbiega niczym od instalacji standardowego modułu, istnieje także możliwość wykorzystania API prosto od firmy MaxMind lub ich bazy danych.

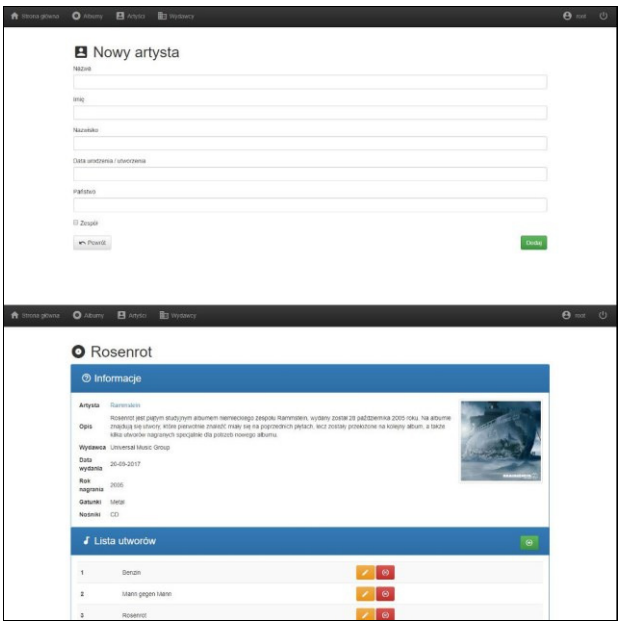
Bazę GeoIP wykorzystuje wiele rozbudowanych aplikacji na czele z Facebook.com. Stworzenie specjalnego modułu dla języka Laravel nie było na pewno konieczne, gdyż API firmy MaxMind sprawnie działa z tym narzędziem, to daje ono większy wybór sposobów i stopnia zaawansowania, w jakim deweloper chce z tej funkcjonalności korzystać – API daje dużo więcej możliwości.

5. Prezentacja rezultatów badań

Wygląd aplikacji napisanej w narzędziu Laravel (analogiczny w każdym z frameworków) przedstawiają Rys. 1 oraz Rys. 2.



Rys.1. Wygląd listy wydawców w aplikacji testowej.



Rys.2. Wygląd formularza dodawania nowego artysty oraz szczegółów albumu w aplikacji testowej

Porównanie stworzonych aplikacji zawiera analizę ich możliwości oraz wydajności. Badane aspekty zostały przedstawione w postaci tabeli, gdzie widać zestawienie trzech wykorzystanych w badaniu frameworków.

5.1. Analiza możliwościowa

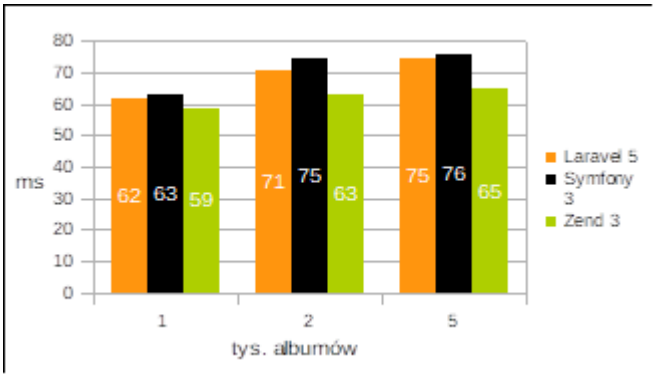
Analizę możliwościową przeprowadzono podczas tworzenia aplikacji testowych. Zaobserwowane aspekty w pracy z poszczególnymi narzędziami zebrano w postaci tabeli i przedstawiono w Tabeli 1. Ukazuje ona analizę czysto programistyczną, skupiając się na wspólnych czynnikach opisywanych frameworków i odpowiednio je opisując.

Tabela 1. Zestawienie aspektów w pracy z poszczególnymi narzędziami języka PHP.

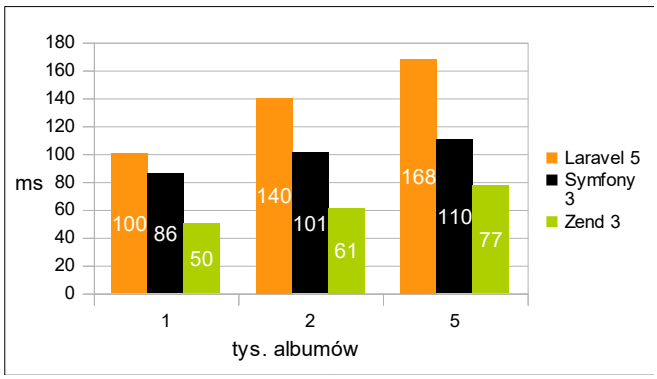
		Laravel 5	Symfony 3	Zend 3
Kontroler	Routing akcji	w oddzielnym pliku	w oddzielnym pliku, w kontrolerze jako adnotacje	w oddzielnym pliku
	Gotowy mechanizm tworzenia kontrolera	rozbudowany, z gotowymi metodami, dla wybranego modelu danych	rozbudowany z wieloma opcjami dla każdego aspektu, szybki z domyślnymi opcjami	nie
Widok	Format	.blade	.twig	.phtml
	Gotowy mechanizm tworzenia widoków	nie	tak	nie
	Zagnież-dżanie widoków	tak	tak	tak
	Dziedziczenie widoków	tak	tak	tak
Kolekcje danych	ORM	Eloquent	Doctrine	nie
	Gotowy mechanizm obsługi relacji pomiędzy tabelami	tak, lecz bez wspomagania relacji wiele do wielu, które należy definiować własnoręcznie w modelu	tak, tworzony jest schemat encji wraz z relacjami a na jego podstawie tabela i model dla obiektów w aplikacji	nie
Baza danych	Obsługiwane systemy baz danych	MSBI, MongoDB, ,MySQL, MariaDB, PgSQL, Redis,SQLite	MS BI MongoDB MySQL Oracle PgSQL MariaDB	MariaDB MySQL MSSQL Oracle PgSQL SQLite FireBird
	Gotowe mechanizmy migracyjne	tak	tak	nie
Formularze	Gotowy mechanizm generowania formularzy	tak, dostępny dopiero po instalacji modułu	tak	tak
	Implementacja z poziomu kontrolera	tak	nie	nie
	Implementacja z poziomu widoku	tak	tak	tak
Poziom trudności tworzenia aplikacji	Znajomość języka PHP	znajomość programowania obiektowego, podstawy pracy z obiektami	znajomość programowania obiektowego, podstawy pracy z obiektami	znajomość programowania obiektowego, podstawy pracy z obiektami
	Ilość gotowych mechanizmów	duża, wszystkie od razu w pełni gotowe do użycia	bardzo duża, jednak wiele jest jedynie rozszerzeniem tych najpopu-larniej-szych	mała
	Instalowanie nowych modułów	wprowadzenie zmian w dwóch plikach konfiguracyjnych	wprowadzenie zmian w dwóch plikach konfiguracyjnych	wprowadzanie zmian w wielu plikach konfiguracyjnych
	Szybkość tworzenia aplikacji w porównaniu z pozostałymi narzędziami	najszybszy	mozolna praca z plikami konfiguracyjnymi, duża swoboda w tworzeniu aplikacji ale kosztem czasu	najwolniejsza, wiele rzeczy trzeba tworzyć samemu, brak gotowych komponentów
	Mechanizm serwerowy do testowania aplikacji	tak	tak	tak

5.2. Analiza wydajnościowa

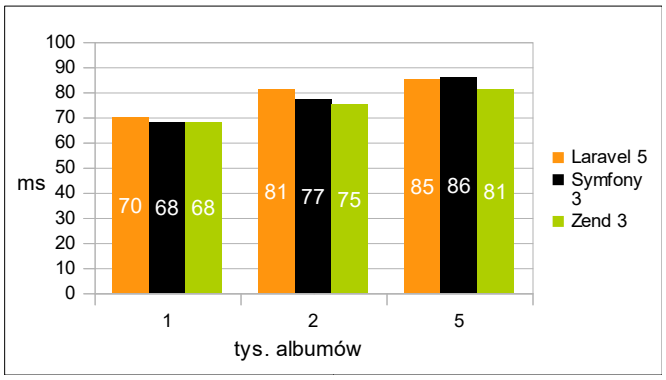
Analizę wydajnościową przedstawiono w postaci wykresów opisujących najważniejsze badane aspekty. Każdy z nich zawiera zestawienie wszystkich badanych narzędzi na tle trzech zestawów danych. W badaniach zwrócono uwagę na czas, opisany na wykresach w milisekundach (ms).



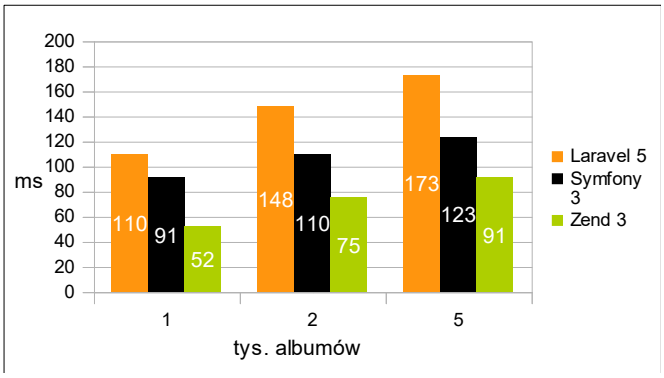
Rys.3. Czas dostępu do bazy danych.



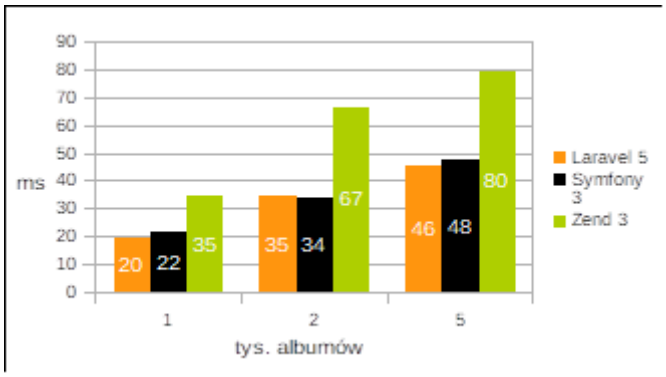
Rys.4. Czas odczytu z bazy danych.



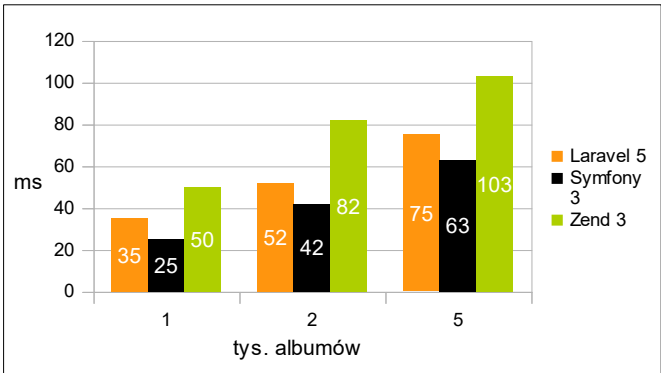
Rys.5. Czas wyszukiwania pojedynczego albumu w bazie danych.



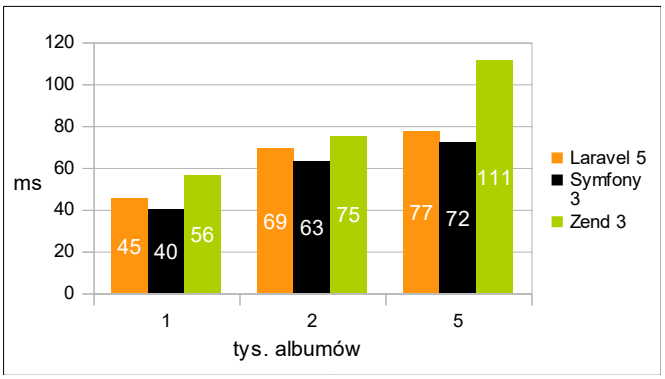
Rys.6. Czas filtrowania albumów w bazie danych.



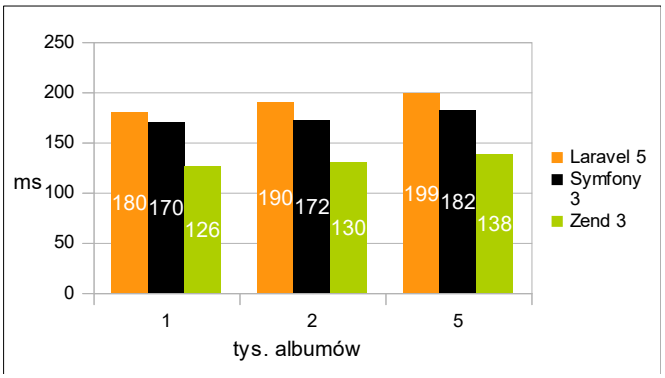
Rys.7. Czas wyszukiwania odpowiednich elementów w pobranej kolekcji danych w bazie.



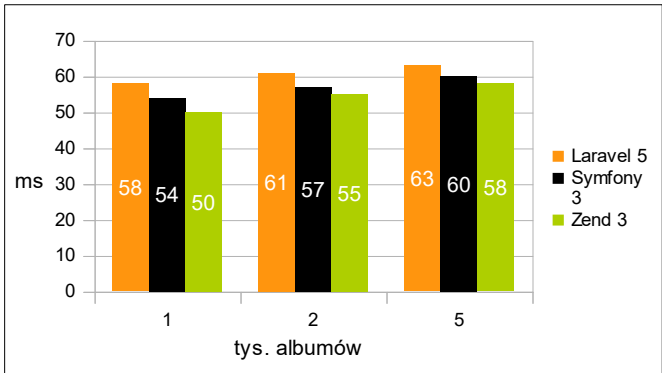
Rys.8. Czas filtrowania odpowiednich elementów w pobranej kolekcji danych w bazie.



Rys.9. Czas generowania kolekcji danych złożonych z wielu encji.



Rys.10. Czas generowania strony.



Rys.11. Czas ładowania plików graficznych.

5.3. Podsumowanie

Na podstawie powyższych analiz można stwierdzić, że Laravel jest narzędziem zdecydowanie najbardziej przystępnym dla programisty. Zarówno Symfony jak i Zend wymagają dużego doświadczenia w tworzeniu aplikacji PHP oraz skupiają się w szczególności na plikach konfiguracyjnych co dla mniej wymagającego dewelopera może okazać się zbyt przytłaczające. Najmłodsze z powyższych narzędzi ogranicza korzystanie z plików konfiguracyjnych do minimum, wymagając jedynie podawania nazw modułów przy ich instalacji czy też podania konfiguracji bazy danych w specjalnym pliku – wszystko inne narzędzie wykona samodzielnie. Także routing okazuje się bardzo intuicyjny i, w przeciwieństwie do Zend, łatwy w definicji. Mechanizm ORM Eloquent pozwala programiście zdefiniować schemat bazy danych i na jego podstawie tworzy encje oraz wspiera późniejsze ich migracje – jedynym mankamentem w tym przypadku są trudności w aktualizacji bazy, co w ORM Doctrine przebiega bezproblemowo. Format .blade, na którym oparte są widoki w Laravel jest zbliżony do .twig używanego w Symfony, posiada jednak bardziej zbliżone do natywnego języka PHP odwołania do obiektów i instrukcji.

Wydajność to w wielu przypadkach najistotniejsza kwestia, przez pryzmat której deweloper wybiera narzędzie to tworzenia internetowej aplikacji. W tym aspekcie Laravel wypadł zdecydowanie najsłabiej ze wszystkich porównywanych frameworków – w większości badań osiągał najdłuższe czasy ładowania, co pokazuje, że system obsługi kolekcji danych Eloquent oraz mechanizm widoków .blade to w porównaniu chociażby z Symfonowym Doctrine oraz .twig mechanizmy przyjazne w obsłudze kosztem szybkości działania aplikacji. Czasy w powyższych badaniach oscylowały w granicach 50-200 milisekund, więc nie są to znacząco odczuwalne przez użytkownika różnice. W bardziej rozbudowanych aplikacjach, posiadających zdecydowanie więcej niż 5 tysięcy wpisów w jednej tabeli i wielu relacji do niej, dysproporcja czasowa byłaby jednak dużo bardziej widoczna i w znaczącym stopniu ograniczała wybór Laravel jako narzędzie ich pisania.

6. Wnioski

Laravel to bardzo dynamicznie rozwijające się narzędzie. Dzięki swojej przystępności szybko zyskał popularność i zbudował społeczność, która wspiera twórców w ciągłym ulepszaniu ich produktu, między innymi poprzez moduły, które każdy użytkownik Laravel może używać na mocy otwartej licencji frameworka i wszystkich jego komponentów.

Jako narzędzie do tworzenia aplikacji internetowych wspiera wiele technologii, pozostawiając programistom pełną swobodę w ich wyborze. Sugerowane przez twórców technologie są natomiast bogato udokumentowane i w większości przypadków wystarczające dla bardziej wymagających deweloperów. Można to zaobserwować na przykładzie przeglądu literatury, gdzie przedstawiono kilka innowacyjnych pomysłów bazujących na Laravel.

Framework posiada wiele dodatkowych funkcji. W łatwy sposób pozwala na tworzenie API i integrację z innymi narzędziami, dodatkowo posiada swoisty miniframework zaprojektowany wyłącznie do API - LUMEN. Wsparcie firmy Amazon modułem bazy GeoIP daje przewagę nad innymi narzędziami PHP – Laravel jako jedyny posiada dedykowany mechanizm dający większy wybór sposobów i stopnia zaawansowania, w jakim programista chce z tej funkcjonalności korzystać.

W badaniu wydajności Laravel niestety nie wypadł najlepiej. Dwa pozostałe badane narzędzia – Symfony i Zend w większości aspektów były wydajniejsze, w tym Zend blisko dwukrotnie. Przyjazny w wykorzystaniu ORM Eloquent okazał się znacznie wolniejszy nich Doctrine zarówno w operacjach na bazie danych jak i lokalnych kolekcjach obiektów. Ogranicza to w znacznym stopniu wybór Laravel jako narzędzie do tworzenia zaawansowanych aplikacji internetowych takich jak Facebook czy YouTube. Nie zmienia to faktu, że jest to bardzo dobre narzędzie, dynamicznie rozwijane przez bogatą społeczność. Pomimo najsłabszej spośród badanych narzędzi wydajności wydaje się odpowiedni dla większości mało wymagających aplikacji, w głównej mierzę dzięki przystępności i bogatej dokumentacji.

Literatura

[1] M. Doyle, J. Walden, “An Empirical Study of the Evolution of PHP Web Application Security”, Department of Computer Science Northern Kentucky University Highland Heights, 2011, s. 4.

[2] A. Zurkiewicz, M. Miłosz, “SELECTING A PHP FRAMEWORK FOR A WEB APPLICATION PROJECT - THE METHOD AND CASE STUDY”, Conference: 9th international Technology, Education and Development Conference, s. 1712

[3] L. Putra, Michael, Yudishtira, “Design and Implementation of Web Based Home Electrical Appliance Monitoring, Diagnosing, and Controlling System”, Procedia Computer Science volume 59, 2015, s. 36

[4] M. Żyłowska, A. Wyszyńska, E. K. Jaguszyn-Krynicka, “Defensyny - peptydy. O aktywności przeciwbakteryjnej”, Instytut Mikrobiologii Uniwersytetu Warszawskiego, Zakład Genetyki Bakterii, marzec 2011, s. 223

[5] E. A. Gomez, P. Giraldo, S. Orduz, “InverPep: A database of invertebrate antimicrobial peptides”, Journal of Global Antimicrobial Resistance volume 8, marzec 2017, s.13 - 17

[6] H. R. Yu, “Design and implementation of web based on Laravel framework”,ACSR-Advances in Comptuer Science Research valume 6, 2015, s. 302

[7] A. Lathifah, T. Aris, „Sentinel Web: Implementation of Laravel Framework in Web Based Temperature and Humidity

Monitoring System", 2nd Int. Conference on Information Technology, Computer and Electrical Engineering, 2015, s.48

[8] http://php.net/manual/pl/preface.php [26.10.2017]

[9] http://ciencias.medellin.unal.edu.co/gruposdeinvestigacion/prospeccionydisenobiomoleculas/InverPep/public/home_en [27.10.2017]

[10] https://laravelcollective.com [26.10.2017]

[11] http://github.com/barryvdh/laravel-debugbar [26.10.2017]

[12] http://maatwebsite.nl/laravel-excel [26.10.2017]

[13] https://packalyst.com/packages/package/jenssegers/agent [26.10.2017]

[14] http://lyften.com/projects/laravel-geoip/doc/ [26.10.2017]